Intruders with Caps

Siva Anantharaman, Paliath Narendran, Michaël Rusinowitch

UNIVERSITE D'ORLEANS

D'INFORMATIQUE
LIFO
FONDAMENTALE
D'ORLEANS

4 rue Léonard de Vinci
BP 6759
F-45067 Orléans Cedex 2
FRANCE
http://www.univ-orleans.fr/lifo

Rapport de Recherche

Intruders with Caps

S. ANANTHARAMAN, LIFO, Orléans (Fr.)
P. NARENDRAN, SUNY at Albany-NY (USA)
M. RUSINOWITCH, LORIA, Nancy (Fr.)

Rapport N^o **2007-02**

Intruders with Caps

Siva Anantharaman[1] and Paliath Narendran[2] and Michael Rusinowitch[3]

[1] LIFO - Universit d'Orleans (France), e-mail: siva@univ-orleans.fr
[2] University at Albany–SUNY (USA), e-mail: dran@cs.albany.edu
[3] Loria-INRIA Lorraine, Nancy (France), e-mail: rusi@loria.fr

Abstract

In the analysis of cryptographic protocols, a *treacherous* set of terms is one from which an intruder can get access to what was intended to be secret, by adding on to the top of a sequence of elements of this set, a *cap* formed of symbols legally part of his/her knowledge. In this paper, we give sufficient conditions on the rewrite system modeling the intruder's abilities, such as using encryption and decryption functions, to ensure that it is decidable if such caps exist. The following classes of intruder systems are studied: linear, dwindling, Δ-strong, and optimally reducing; and depending on the class considered, the cap problem ("find a cap for a given set of terms") is shown respectively to be in P, NP-complete, decidable, and undecidable.

1 Introduction

Cryptography has been applied to render communications secure over an insecure network for many years. However, the underlying difficulties in properly designing cryptographic protocols are reflected by repeated discovery of *logical* bugs in these protocols. As an attempt to solve the problem, there has been a sustained and successful effort to devise formal methods for specifying and verifying the security goals of cryptoprotocols. Various symbolic approaches have been proposed to represent protocols and reason about them, and to attempt to verify security properties such as confidentiality and authenticity, or to discover bugs. Such approaches include process algebra, model-checking, modal logics, equational reasoning, and resolution theorem-proving (e.g. [23, 2, 9, 4]).

In particular, string rewrite systems have provided one of the first formal treatments of security protocol analysis [15], by modeling encryption and decryption as abstract operators. In such a setting, the secrecy property – i.e., whether a message can be deduced by the intruder from observed communications – can be reduced to the so-called *extended word problems*. The approach has been generalized to more realistic protocols by employing term rewrite rules [16, 12, 21], in particular modeling the capabilities of the intruder in terms of a convergent term rewrite system (*TRS*, for short); more elaborate primitives can be obtained that way. In the analysis of cryptographic protocols using such an approach, the *general cap problem* (that we shall define shortly), formally models the possibility that a passive intruder gets hold of a secret m, by using – and possibly re-using – some of the non-public terms that (s)he captures, e.g., by eavesdropping, during a given protocol session. The issue addressed in this paper is how general the (convergent) TRS modeling

the intruder's capabilities can be, so as to get tractable decision procedures for solving this problem.

This paper is structured as follows: In Section 2, after some preliminaries. we formally define the general cap problem, as well as a simpler variant called just the cap problem. Section 3 shows in some detail how these cap problems can be applied to the formal security analysis of protocols. In Section 4, the cap problem is shown to be decidable for optimally reducing string rewrite systems. Section 5 studies the cap problem for (convergent) dwindling TRS R; it is shown to be decidable in polynomial time; if the TRS R is assumed left-linear in addition, then we show that the set of all irreducible treacherous terms is a regular tree language. Section 6 establishes the undecidability of the cap problem for (convergent) linear, optimally reducing TRS, by reduction from the halting problem for 2-counter machines. In Section 7, we turn our attention to the general cap problem and show that it is NP-complete for special or dwindling TRS. The general cap problem is then studied with respect to a class of rewrite systems R called Δ-strong, that extends the class of dwindling systems, and the decidability of the general cap problem wrt such systems R is shown (Section 8); possible applications are that of modeling homomorphic encryption and the blind signature protocol; a slightly more general notion, called $\omega\Delta$-strong, is developed in a subsection. The concluding section compares our work with related works, and presents a few possible directions for future work. Appendix-I shows that the cap problem may be decidable for intruder systems R for which semantic unification is undecidable; and Appendix-II shows that the complexity of the general cap problem is non-primitive recursive, for Δ-strong intruder theories.

2 Notation and Preliminaries

We assume that the reader is familiar with the well-known notions of terms, rewrite rules and rewrite systems over a given (ranked) signature Σ, and a (possibly infinite) set of variables $\mathcal{X}$. For any term t, the set of all its positions will be denoted as $Pos(t)$; if $q \in Pos(t)$ then $t|_q$ will denote the subterm of t at position q; and following Huet [18], the term obtained from t by replacing the subterm $t|_q$ by a term t' will be denoted as $t[q \leftarrow t']$. A similar notation will also be used for the substitution of variables in t by terms. The notions of reduction and of normalization of a term by a rewrite system are assumed familiar too, as well as those of termination and of confluence of the reduction relation defined by such a system on terms. A rewrite system is said to be *convergent* iff the reduction relation it defines on the set of terms is terminating and confluent.

The Cap Problems: Let R be any convergent TRS over some ranked signature Σ and a variable set $\mathcal{X}$. We assume a ground constant $m \in \Sigma$, referred to as the *secret,* and a subset G of $\Sigma \setminus \{m\}$ referred to as the *intruder repertoire* or *public* symbols. It is assumed that G contains all the root symbols of the left hand sides of all the rules in R and at least one constant, and also that m appears nowhere in the rules of R. ("R *is free from* m"). Symbols which are not in the intruder repertoire are often referred to as *private* symbols. A term that contains only public symbols (and variables) will be said to be a public term; it is said to be private, or non-public, otherwise.

We then extend the signature by adding a set $\{\diamond, \diamond', \diamond'', \cdots\}$, of special variables referred to as *hole variables*, or just *holes*; the symbols $\diamond, \diamond_1, \diamond_2, \cdots$ (with or without primes) will

be used to designate any of the hole variables. A *cap,* or a *cap-term,* is then defined as a public term such that the only variables in it are hole variables. Caps are often represented as $t(\diamond_1, \ldots, \diamond_n)$, where the $\diamond_i, 1 \leq i \leq n$, are the distinct hole variables of t, *each of which may occur more than once*. A cap with *exactly one* hole variable occurrence, at a position q, is often more conveniently denoted as $t[\,]_q$. The problem referred to in this paper as the *general cap problem*, is the following:

Instance: A convergent TRS R with the properties mentioned above, an intruder repertoire G, and a finite set S of *non-public* ground terms over Σ, at least one of which contains the secret m.

Question: Is there a cap $t(\diamond_1, \ldots, \diamond_n)$ over the intruder repertoire G, such that a term $t[\diamond_1 \leftarrow s_1, \cdots, \diamond_n \leftarrow s_n]$, with the $s_i \in S$ (not necessarily all distinct), can be R-reduced to m?

If this question admits a positive answer, the multiset $\{s_1, \cdots, s_n\}$, as well as the set S itself, will be said to be *treacherous*, wrt R. The following simpler version of the problem, where the cap has just *one* hole variable occurrence, is referred to as the *cap problem* in the sequel:

Instance': A convergent TRS R with the properties mentioned above, an intruder repertoire G. and a ground term s containing the secret m.

Question': Is there a cap $t[\,]_q$ over the intruder repertoire G, such that the term $t[q \leftarrow s]$ reduces to m?

This simpler version models the possibility that the intruder gets hold of m without re-using any of the intermediary terms captured during a protocol session. The general cap problem will be studied only in the later sections of this paper. We shall first study the (simpler version of the) cap problem, for the following classes of rewrite systems R: string rewrite systems that are either *special* or *optimally reducing,* and term rewrite systems that are either *dwindling* or *optimally reducing.* These notions are formally defined as follows:

i) R is *special* (or *pure*) iff the rhs of every rule in R is a variable.

ii) R is *dwindling* iff, for every rule $l \rightarrow r \in R$, r is a *proper* subterm of l.

iii) R is *optimally reducing* iff, for every $l \rightarrow r \in R$, and for any substitution θ on $\mathcal{X}$ for which $\theta(r)$ is reducible, there is a *proper* subterm s of l such that $\theta(s)$ is reducible[1].

The reason for considering these classes is that, in the formal models of several protocols, term rewrite systems that model the intruder capabilities often belong to these classes. Note that the above three notions are decreasingly restrictive. It is decidable whether a given TRS R is optimally reducing: a non-deterministic polynomial time decision procedure is given in [20]; it is also shown there that unification modulo a convergent optimally reducing TRS R is decidable, by innermost narrowing.

Recall that a string rewrite system over an alphabet Σ can be seen as the particular case of TRS where the symbols in Σ are all of rank 1. For redactional reasons, we shall

[1]This notion was first introduced in [20], and has been extended recently in [11].

agree to view, in the sequel, any string u over Σ as a term over one variable derived from the *reversed* string of u; i.e., if $g, h \in \Sigma$ the string gh will be seen as the term $h(g(x))$.

That agreed upon, the above three notions on TRS can be reformulated for string rewrite systems, as follows: a string rewrite system T is special iff the rhs of each rule in T is the empty string; T is dwindling iff, for every rule $l \to r$ in T, the rhs r is a *proper prefix* of its lhs l; and T is optimally reducing, iff the following holds:

> For every rule $ub \to v$ in T, with $u, v \in \Sigma^*$, and $b \in \Sigma$, and for all strings z, if zv is reducible then so is zu.

3 Security Analysis and the Cap Problem

In this section, we show briefly how the cap problem appears naturally in the formal analysis of protocol (or system) security. In many approaches, the protocol (or the system's capacity) is modeled either in terms of suitable Horn clauses (e.g. [23, 4]), or as a multiset rewrite system (e.g. [16]); and the intruder capabilities are modeled as a rewrite system which is in general special (e.g. [21]).

3.1 String Case

We shall consider first the approach initiated by Dolev and Yao [15] in 1983, for the security analysis of some two-party public key protocols. In this model honest parties are stateless: the messages transmitted by a party at every step of the protocol are a function of the message they just received and there is no control on the received messages. More precisely, the protocols are represented as sequences of strings on an alphabet of unary operators. The i-th string is added as a cap (i.e., applied as a function) to the i-th received message, in order to determine the next message that is transmitted in the network.

On the other hand, the adversary is assumed to have total control of the network. In particular (s)he can interfere with the concurrent execution of an arbitrary number of protocol executions. The goal of the adversary is to recover some message M. Under this execution model the adversary has access to some encryption/decryption (also append/delete namestamp) functions – that (s)he can apply to the circulating messages of his/her choice. These functions are modeled as a convergent system of special string rewrite rules R. Since the intruder can impersonate honest parties (s)he can also apply protocol strings to messages (s)he has in his/her possession. It can be shown that the set of messages the intruder can generate in such a manner is a regular language.

The 2-party protocols considered in [15] were defined formally as insecure if and only if a certain initial message exchanged between the parties can be normalized to the empty string by the intruder rewrite system R, by successively adding on caps to the initial message, where each cap is built by using the intruder capabilities. It was observed by Book and Otto (e.g., [7]), that the main technical result behind the Dolev-Yao result can be formulated as follows:

> Let R be a convergent special string rewrite system. Then for any regular language L, the set of all descendants of strings from L, i.e., $\{x \mid \exists y \in L : \ y \to^* x\}$, is a regular language. A non-deterministic finite automaton (NFA) accepting this language can be constructed in time polynomial in the total size of R and the size of the NFA.

3.2 Term Case

When considering cryptographic protocols defined with operators of arity greater than 1, the extension of cap problems from strings to terms is still relevant for security analysis. Our approach is basically motivated by the logical approach to security where protocol rules, from the intruder's point of view, are modeled using Horn clauses (e.g. [23]) or deduction rules (e.g. [22]).

Consider the following elementary ping-pong protocol introduced in [15]:

$$\begin{array}{ll} A \to B: & A, B, \{M\}_{kb} \\ B \to A: & B, A, \{M\}_{ka} \end{array}$$

An intruder impersonating B can mount the following easy attack:

$$\begin{array}{ll} A \to I(B): & A, B, \{M\}_{kb} \\ I \to B: & I, B, \{M\}_{kb} \\ B \to I: & I, B, \{M\}_{ki} \end{array}$$

This can be expressed using encryption and decryption functions e and d respectively and with kb and ki as the keys of B and I respectively. The intruder's initial knowledge includes b, i, kb, ki. At the end of the protocol I gets $e(d(e(m, kb)), kb), ki)$. Modulo the convergent term rewrite system $\mathcal{I}$, consisting of the following rules:

$$d(e(y, u), u) \to y$$

$$e(d(y, u), u) \to y$$

this is equivalent to $e(m, ki)$. However, this still has not shown that the intruder can get hold of m. For that we have to find a suitable cap for $e(m, ki)$ so that the capped term will normalize to m.

For the case where messages are terms, we shall assume that the intruder has been able to capture some messages from the protocol (we do not study further how the intruder has interacted with the protocol to get these messages). We will rather focus on the problem of finding caps, and on its complexity, for given intruder knowledge, and given secret. The cap problem is equivalent to what is considered in the literature as a security problem in presence of a *passive intruder*. (It is sometimes referred to as the deduction problem, e.g., [1]).

4 The Cap Problem in the String case

As we mentioned earlier (Section 3.1), in the string case the functioning of the protocol is modeled as a regular word grammar, over some given alphabet Σ; the intruder is assumed active: (s)he is allowed to use the protocol rules for capturing the secret; and the system R modeling the intruder capabilities is assumed convergent. We have then the following extension of the cap problem, which is known to be equivalent to protocol insecurity in this case (of an active Dolev-Yao intruder):

Proposition 1 *The following problem is decidable:*

Instance: *An optimally reducing convergent string rewrite system R over an alphabet Σ, an R-irreducible string α, and a regular language $L \subseteq \Sigma^*$.*

Question: *Is there a string $\beta \in L$ such that $\alpha\beta \rightarrow_R^* \lambda$ (the empty string) ?*

Proof: Define $L' = \alpha.L$; then there exists a $\beta \in L$ such that $\alpha.\beta \rightarrow_R^* \lambda$, iff $\lambda \in R^!(L') =$ the set of all R-irreducible descendants of L'. The above proposition can thus be derived by showing that, for *any* regular language L over Σ, the set $R^!(L)$ of all R-irreducible descendants of L is a regular language too. This is done in the following 3 lemmas.

Lemma 1 *Let R be an optimally reducing convergent string rewrite system over the alphabet Σ. Then every congruence class modulo R is a deterministic context-free language.*

Proof: A deterministic push-down automaton (DPDA) can be constructed for each congruence class. We describe the DPDA in terms of the following transition system on tuples from $\Sigma^* \times \Sigma$. The first component of the tuple has the contents of the stack from bottom to top, and the second component has the current tape symbol. The main loop invariant is that the stack contains an irreducible string – in particular the normal form of the string read so far:

$$\begin{array}{ll} (w, a) \mapsto (wa, \epsilon) & \text{if no suffix of } wa \text{ is a redex.} \\ (xl', a) \mapsto (xr, \epsilon) & \text{if } l'a \rightarrow r \text{ is a rule} \end{array}$$

Checking the condition – whether attaching the tape symbol to the stack contents will create a redex – can be incorporated into the finite control of the DPDA; e.g., by building a trie of all the left-hand sides. □

Lemma 2 *Let R be an optimally reducing convergent string rewrite system over the alphabet Σ and let $\#$ be a symbol not in Σ. Then the language*

$$\{x\#y \mid x, y \in \Sigma^*,\ y^{rev} \text{ is } R\text{-irreducible},\ x \rightarrow_R^! y^{rev}\}$$

is context-free. (y^{rev} is the reverse string of y.)

Proof: The main idea is the same as in the proof of the previous lemma. We construct a DPDA that will scan an input string of the form $x\#y$ from left to right, and will have the normal form of x in the stack when it reaches the tape cell that contains the $\#$ symbol. From then on, the machine pops the stack when the symbol at its top and the tape symbol agree. □

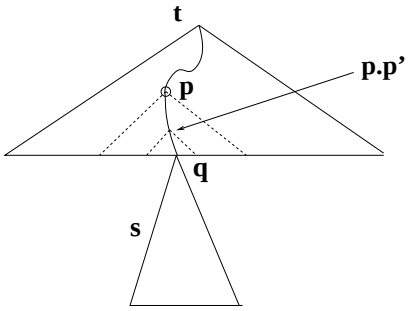
Lemma 3 *Let R be an optimally reducing convergent string rewrite system over the alphabet Σ, and let $L \subseteq \Sigma^*$ be a regular language. Then the language $R^!(L) = \{u \mid \exists v \in L : v \rightarrow^! u\}$ is a regular language.*

Proof: This follows from the preceding lemma, and the proof is essentially the same as that of Theorem 2.5 in [6]. □

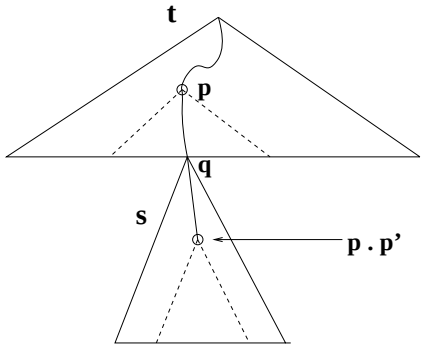
5 Deciding the Cap Problem for dwindling TRS

We give here a recursive algorithm for solving the cap problem, under the assumption that the given TRS R is dwindling. With the notation of Section 2, we can assume without loss of generality that both the given term s (containing the secret m) and the cap to be found are R-irreducible. Suppose $t[\,]_q$ is a minimal cap, that allows one to deduce m, and let $t' = t[q \leftarrow s]$. Let p be the innermost position where t' is reducible by a rule, say $l \to r$. Clearly we have $p \prec q$ (for the prefix-ordering '$\preceq$' on positions). Thus $t'|_p = \sigma(l)$ for some substitution σ. Let p' be a position in l such that $l|_{p'} = r$; then $\sigma(l)|_{p'} = \sigma(r)$, and $t'|_{p \cdot p'} = \sigma(r)$. Now two cases have to be considered.

Case (i) $p \cdot p' \preceq q$: In this case $t'|_p$ reduces to $t'|_{p \cdot p'}$. Hence $t[p \leftarrow t|_{p \cdot p'}]_q$ will be a smaller cap (with a hole at some position above q), so this case need not be considered. The sub-case where $p \cdot p' = q$ for every possible redex is one where there is no cap, and one exits with failure.



Case (ii) $q \prec p \cdot p'$: Let $q = p \cdot q'$ and $p' = q' \cdot q''$. We have here: $q' \prec p'$, $\sigma(l|_{q'}) = s$. And $\sigma(l)$ reduces to $s|_{q''}$ which is a proper subterm of s. Thus s allows us to deduce m if and only if $s|_{q''}$ does so too.



Note that the case where $p \cdot p'$ and q are incomparable need not be considered since the cap is assumed irreducible (and cannot contain any occurrence of m). We thus derive a procedure for checking whether a term is treacherous:

1. If $s = m$ RETURN **true**;
   else non-deterministically choose a rule $l \to r$ and a proper subterm l' of l that is a *proper* superterm of r; let $l' = l|_q$.
2. Let $\theta = mgu(s =^? l')$.
3. If $\theta(l[q \leftarrow \diamond])$ has symbols which are not in the intruder repertoire, or if $\theta(r)$ does *not* contain m, then **fail**;
   else set $s := \theta(r)$ and GOTO 1.

This can also be done in a bottom-up (dynamic programming-like) way. Clearly m itself is treacherous. Now suppose all proper subterms of s have been tested for treachery and the results recorded. This is tantamount to annotating each subterm by T or F depending on whether it is treacherous or not. Testing whether s itself is treacherous can then be done by modifying Step 3 above to:

3bis. If $\theta(l[q \leftarrow \diamond])$ has symbols which are not in the intruder repertoire, **fail**;
else check whether $s := \theta(r)$ is treacherous: if **yes** RETURN **true** else **fail**.

Making this deterministic requires that each such l' be tried in Step 1. This could take $O(|R||s|)$ time in the worst case, where $|R|$ is the total size of the term rewrite system. Thus the total complexity is $O(|R||s|^2)$.

5.1 Case of Left-linear dwindling TRS: a Regularity Result

When the TRS is also left-linear, we can derive a more precise result about the set of all irreducible treacherous terms. Observe first that the linearity of the lhs of the rules in R allows us to reformulate the above algorithm, as follows:

1'. Non-deterministically choose a rule $l \rightarrow r$ and a *proper* subterm $l' = l|_q$ of l, that is a *proper* superterm of r, with the additional property that $(l[q \leftarrow \diamond])$ has all its symbols inside the intruder repertoire.

2'. Let $\theta = mgu(s =^? l')$.

3'. Set $s := \theta(r)$, and GOTO 1'.

Proposition 2 *Let R be a left-linear, dwindling and convergent TRS. Then the set of all* irreducible *treacherous terms, wrt R, is a regular tree language.*

Before proving this proposition, let us observe that the hypothesis of *left-linearity* is essential, as the following example shows: consider the special TRS formed of the unique rule $f(g(x,x,y)) \rightarrow y$ where f is public; then, clearly $g(t_1,t_2,m)$ is treacherous if and only if $t_1 = t_2$. Note also that the hypothesis of *irreducibility* is also needed, as is seen with the example of the string rewrite system with a single rule $f\,g \rightarrow \lambda$; the set of *all* treacherous terms here is non-regular, since its intersection with f^*g^* is the language $\{f^n\,g^m \mid n \geq m\}$; but the set of all *irreducible* treacherous terms is $\{f^n \mid n \geq 0\} = f^*$.

The above proposition is proved via the following lines of reasoning:

(i) we construct a regular tree grammar $\mathcal{G}$ that generates a subset of the set of all treacherous terms wrt R, which includes all irreducible treacherous terms;

(ii) since R is assumed left-linear, the set $IRR(R)$ of all ground terms in R-normal form is known to be a regular tree language (cf. e.g., [17]);

(iii) The set of all irreducible treacherous terms is then obtained as the intersection of the language of $\mathcal{G}$ with $IRR(R)$.

Proof of Proposition 2: We construct a regular tree grammar that generates a subset of all terms that are treacherous wrt R. The underlying idea comes from the simplified version of the algorithm, given above, for finding minimal caps.

Let $\{l_i \to r_i \mid 1 \leq i \leq N\}$ be the set of all rules of the given dwindling TRS R. Let k be the maximum depth of left-hand sides of R, and let Π be the set of all positional sequences of length k or less; i.e., $\Pi = \bigcup_{i=0}^{k} \{1, \ldots, \alpha\}^i$ where α is the maximum arity of the function symbols in R.

Let T be the set of non-variable proper subterms of the left-hand-sides of R; i.e., $T = \{\, l_i|_u \mid l_i \in lhs(R),\ u \neq \epsilon,\ u \text{ is not a variable position of } l_i \,\}$. And let

$$\begin{aligned} T_1 &= \{\, l_i|_u \mid l_i|_u \in T,\ (l_i[u \leftarrow \diamond]) \in \mathcal{T}(G \cup \{\diamond\}, \mathcal{X}) \quad \text{and} \\ T_2 &= \{\, l_i|_u \mid l_i|_u \in T_1,\ r_i \text{ is a proper subterm of } l_i|_u \}. \end{aligned}$$

In other words, T_2 consists of the subterms $l_i|_u$ at non-root positions of the left-hand sides of R such that

– $l_i|_u$ is also a *proper superterm* of r_i,

– all the symbols of the term $(l_i[u \leftarrow \diamond])$ are in the intruder repertoire.

We now define the production rules of the regular tree grammar $\mathcal{G}$. The non-terminals of $\mathcal{G}$ are defined as the elements of the set $\{0,1\} \times \{1, \ldots, N\} \times \Pi \times \Pi \times \{0,1\} \times 2^T$. The first component stands for whether m occurs in the term. The second component stands for the number of the "current rule" in R. The third and fourth are positions in the lhs of the current rule. The fifth denotes whether the current term is a treacherous term. The sixth is a set of terms with the property that the current term generated is an instance of every one of them.

The axioms of the grammar are defined as the non-terminals of the set:

$$\{\, (1, j, v, p, 1, \{l_j|_v\}) \ \mid\ l_j|_v \in T_2,\ r_j = l_j|_{v \cdot p} \,\}$$

The production rules of the tree grammar are grouped into four different types, as indicated below; the symbol '?' and '×' respectively stand for "*don't-know*" and "*don't-care*"; and '×' is allowed to take any value, other than '?', that is appropriate to the context:

Type (i) – the current term is a treacherous term:

$$(1, j, v, p, 1, \Gamma) \ \to\ f(\,(b_1, j_1, v_1, p_1, a_1, \Gamma_1), \ldots, (b_n, j_n, v_n, p_n, a_n, \Gamma_n)\,)$$

for non-empty Γ, provided

1. $f \in \Sigma^{(n)}$, and every term in Γ has f as the root symbol;
2. if Γ contains a term $f(t_1, \ldots, t_n)$, then, for all i, $t_i \in \Gamma_i$, or t_i is a variable;
3. $l_j|_v \in \Gamma \cap T_2$ and $r_j = l_j|_{v \cdot p}$;
4. exactly one of the b_i's is a 1;

5. if $b_k = 1$ and $|p| > 1$, then $j_k = j$, $a_k = ?$, $v_k = v \cdot k$ and $p = k \cdot p_k$;
6. if $b_k = 1$ and $|p| = 1$, then $a_k = 1$.

Type (ii) – the current term is not a treacherous term:

$$(1, j, v, p, 0, \Gamma) \;\rightarrow\; f(\,(b_1, j_1, v_1, p_1, a_1, \Gamma_1), \ldots, (b_n, j_n, v_n, p_n, a_n, \Gamma_n)\,)$$

for non-empty Γ, provided

1. $f \in \Sigma^{(n)}$, and every term in Γ has f as the root symbol;
2. if Γ contains a term $f(t_1, \ldots, t_n)$, then, for all i, $t_i \in \Gamma_i$, or t_i is a variable;
3. $l_j|_v \in \Gamma \cap T$ and $r_j = l_j|_{v \cdot p}$;
4. exactly one of the b_i's is a 1;
5. if $b_k = 1$ and $|p| > 1$, then $j_k = j$, $a_k = ?$, $v_k = v \cdot k$ and $p = k \cdot p_k$;
6. if $b_k = 1$ and $|p| = 1$, then $a_k = 1$.

Type (iii) – the current term does not contain m:

$$(0, \times, \times, \times, \times, \Gamma) \;\rightarrow\; f(\,(0, \times, \times, \times, \times, \Gamma_1), \ldots, (0, \times, \times, \times, \times, \Gamma_n)\,)$$

for non-empty Γ, provided

1. $f \in \Sigma^{(n)}$, and every term in Γ has f as the root symbol;
2. if Γ contains a term $f(t_1, \ldots, t_n)$, then, for all i, $t_i \in \Gamma_i$ or t_i is a variable.

(Note that the only difference between (i) and (ii) is in Condition 3; note also that, in (iii), only the 'instance requirement' – that the current term be an instance of all the terms in Γ – needs to be met.)

Type (iv) – Generation of the ground constants:

$$\begin{array}{rcll} (1, \times, \times, \times, 1, \emptyset) & \rightarrow & m, & \\ (0, \times, \times, \times, 1, \{d\}) & \rightarrow & d, & \text{for any public constant } d, \\ (0, \times, \times, \times, \times, \emptyset) & \rightarrow & c, & \text{for any constant } c \neq m. \end{array}$$

Thanks to our earlier considerations, one can check that the tree grammar $\mathcal{G}$ generates only R-treacherous terms, and that every irreducible R-treacherous term is generated by $\mathcal{G}$. □

Figure 1 illustrates the reasonings above. Consider the following linear TRS:

$$(1)\; f(g(g(g(x)))) \rightarrow g(x)\;, \qquad (2)\; f(g(h(x))) \rightarrow x,$$

with f as the only public symbol. The irreducible treacherous term $g(g(g(h(m))))$ can be generated in the (top-down) sequence shown in the figure. The axioms of the grammar are $(1, 1, 1 \cdot 1, 1, \{g(g(g(X)))\})$ and $(1, 2, 1, 1 \cdot 1, 1, \{g(h(Y))\})$. Note: the 'new' term $g(h(Y))$ can be introduced at the third step because Condition 2 in rules of Types (i), (ii) and (iii) is an "if-then", and <u>not</u> an "iff".

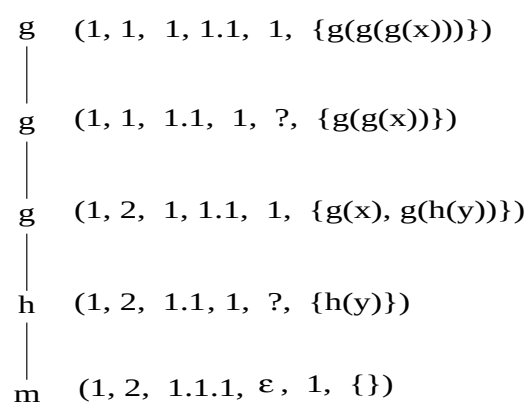


Figure 1: Generating a treacherous term.

6 Undecidability results

Unfortunately the cap problem is undecidable even for optimally reducing, linear, and convergent TRS: we prove that there is a *fixed* optimally reducing, linear, and convergent TRS for which the cap problem is undecidable. The proof is by reduction from the halting problem for 2-counter Minsky machines, that we first present briefly (e.g., as in [3]). A Minsky machine with two counters C_1, C_2 storing non-negative integer values, executes *programs* which are finite lists of *instructions* labeled with the natural numbers from 1 to L, each having one of the following forms, where $0 \leq l \leq (L-1)$, $0 \leq k \leq L$, $k \neq l$, and i is 1 or 2 :

(i) 0: BEGIN

(ii) l: ADD 1 to C_i and GOTO $l+1$;

(iii) l: If $C_i \neq 0$
then SUBTRACT 1 from C_i and GOTO $l+1$;
else GOTO $k \neq 0$;

(iv) L: STOP.

Any such given program $\mathcal{P}$ is assumed to have exactly one instruction 'BEGIN', and one instruction 'STOP'. To every instruction l we associate a state denoted q_l; q_0 (resp. q_L) is defined as the initial (resp. final) state. A *configuration* of such a 2-counter machine, at any given stage of a computation, will be seen as a triple (C_1, q_l, C_2) where l is the (label of the) next instruction to execute, and $C_i, i = 1, 2$, are the current values of the two counters. The following classical result, on the halting problem for such machines, will serve our purposes.

Result: *(Minsky [19])* Let C_1, C_2, C_1', C_2' be any given non-negative integers. Then it is undecidable whether an arbitrarily chosen program $\mathcal{P}$, starting with (C_1, q_0, C_2) as its initial configuration, will halt with (C_1', q_L, C_2') as its final configuration.

We shall encode the halting problem for any program $\mathcal{P}$ of the above type as a cap problem over a rewrite system which is optimally reducing, linear, and convergent. For doing that, any state symbol q_l will be seen as a unary function symbol. In addition we introduce further function symbols f, s of rank 1 and c of rank 3, and constants $m, 0$. The constant m stands for some secret message, 0 encodes the natural integer zero, s encodes

the successor function on integers, and c encodes configuration triples. The following rules (where l, l' stand for suitable instruction labels) do this encoding:

Initial and final configurations (resp. with given k, p, and k', p'):

$$c(s^k(0), q_0(m), s^p(0)), \qquad c(s^{k'}(0), q_L(m), s^{p'}(0))$$

Incrementation of counter 1 or 2:

$$f(c(x, q_l(z), y)) \to c(s(x), q_{l+1}(z), y), \qquad f(c(x, q_l(z), y)) \to c(x, q_{l+1}(z), s(y))$$

Conditional decrementation of counter 1 or 2:

$$f(c(s(x), q_l(z), y)) \to c(x, q_{l+1}(z), y), \qquad f(0, q_l(z), y)) \to c(0, q_{l'}(z), y).$$
$$f(c(x, q_l(z), s(y))) \to c(x, q_{l+1}(z), y), \qquad f(x, q_l(z), 0)) \to c(x, q_{l'}(z), 0).$$

At STOP, release the secret m: $\qquad f(c(s^{k'}(0), q_L(z), s^{p'}(0))) \to z.$

The role played by f is to ensure that this rewrite system is terminating. The cap problem over this rewrite system –which is obviously linear and optimally reducing– with $\{0, f, q_1, \ldots, q_N\}$ as the intruder repertoire, obviously encodes the halting problem for the 2-counter machine programs. We deduce that the cap problem is undecidable even for linear and optimally reducing systems.

7 The General Cap Problem

The definition of the notion of cap with one hole, as given in Section 2, does not allow the intruder to re-use terms. For instance, consider the rewrite system R with a single rule:

$$g(f(x, a), f(y, a)) \to x,$$

where g is in the intruder repertoire, but f and a are not. For the definition of cap with one hole, $f(m, a)$ is not treacherous. But if $f(m, a)$ can be re-used then m can be recovered since $g(f(m, a), f(m, a))$ reduces to m. In other words, there is a (non-linear) cap $t(\diamond) = g(\diamond, \diamond)$ such that $t[\diamond \leftarrow f(m, a)] \to_R^! m$. Also, there could be more than one term containing m that the intruder may be able to use; this explains that the general cap problem allows more general contexts, with more than one hole.

We show now that the general cap problem is NP-complete for dwindling (and convergent) TRSs.

Proposition 3 *The general cap problem is NP-hard, even for special TRS.*

Proof: The proof is by reduction from the 3-colorability problem. Let (V, E) be any arbitrarily given undirected graph. Introduce a function symbol g of rank $|E|+1$, a symbol f of rank 2, and a symbol h of rank 1. Associate a variable x_j to each node v_j in V, and represent every edge $e_i = (v_j, v_k)$ in E (joining the two nodes v_j, v_k in the graph) by the term $t_i = f(x_j, x_k)$; finally let B, G and R be constants that correspond to the 3 colors. We then consider the pure TRS formed of the following single rule:

$$g(t_1, \ldots, t_{|E|}, h(u)) \to u$$

where u is a new variable, not appearing in any of the terms t_i. Assume that g is the only symbol in the intruder repertoire, i.e., all symbols other than g are private. Let $f(B,R)$, $f(R,B)$, $f(G,R)$, $f(R,G)$, $f(G,B)$, $f(B,G)$ and $h(m)$ be the terms known to the intruder. Then it is not hard to see that m can be obtained by the intruder (by plugging in a suitable treacherous set of terms in the cap-term $g(\diamond_1, \cdots, \diamond_{(|E|+1)})$, if and only if the graph can be colored with the 3 colors B, R, G. □

We shall show below that the general cap problem is in NP for dwindling TRS. A few preliminaries are needed for proving that. (They will also be needed farther down, to show that the general cap problem is decidable for the more general Δ-strong and $\omega\Delta$-strong intruder theories, although not in NP.)

7.1 The $\mathcal{I}$-closure of a Set of Terms

Given a finite set S of private ground terms, we define the set of $\mathcal{I}$-constructible terms – referred to as the $\mathcal{I}$-closure $\mathcal{I}(S)$ of S – as follows (the $\mathcal{I}$ refers to the intruder theory):

- $S \subseteq \mathcal{I}(S)$
- If $f^{(p)}$ is a public function symbol and $s_1, \ldots s_p$ are I-constructible terms, then $f(s_1, \ldots s_p) \in \mathcal{I}(S)$
- Nothing else is in $\mathcal{I}(S)$.

In other words, a private ground term t is in $\mathcal{I}(S)$ if and only if either t itself is in S, or the root symbol of t is public and all its top-level subterms are in $\mathcal{I}(S)$. It is not hard to see that $\mathcal{I}(S)$ is a regular tree language for any given finite set S (see the proof of Proposition 7). Define a set of terms $\Gamma = \{t_1, \ldots, t_n\}$ to be $\mathcal{I}$-independent if and only if for all t_i, we have $t_i \notin \mathcal{I}(\Gamma \setminus \{t_i\})$; it is easy to see then, that from every finite set S of terms we can extract an $\mathcal{I}$-independent subset with the same $\mathcal{I}$-closure. A ground substitution θ is $\mathcal{I}$-independent if and only if $\mathcal{R}an(\theta)$ is an $\mathcal{I}$-independent set and $\forall v_i, v_j \in \mathcal{D}om(\theta) : \theta(v_i) = \theta(v_j) \Leftrightarrow v_i = v_j$.

As a direct consequence of the definitions, we get the following: If S is an $\mathcal{I}$–independent set of terms, then a term s is in $\mathcal{I}(S)$ if and only if there is a cap $t(\diamond_1, \ldots, \diamond_n)$ and an $\mathcal{I}$-independent substitution $\theta = [\diamond_1 \leftarrow s_1, \ldots, \diamond_n \leftarrow s_n]$, with $s_i \in S$ for all i, such that $s = \theta(t)$. Proposition 4 and Corollary 1 are easily established too:

Proposition 4 *Let S be a treacherous set of terms and let $t(\diamond_1, \ldots, \diamond_n)$ be a cap for S such that a term $t[\diamond_1 \leftarrow s_1, \cdots, \diamond_n \leftarrow s_n]$, with the $s_i \in S$, can be R-reduced to m. If S' is a $\mathcal{I}$-independent subset of S with the same $\mathcal{I}$-closure, then there is a cap $t'(\diamond'_1, \ldots, \diamond'_k)$ and an $\mathcal{I}$-independent substitution θ with $\mathcal{R}ange(\theta) \subseteq S'$ such that $\theta(t') = t[\diamond_1 \leftarrow s_1, \cdots, \diamond_n \leftarrow s_n]$.*

Corollary 1 *Let S be a treacherous set of terms and let S' be an $\mathcal{I}$-independent subset of S. Then S' is treacherous too. (In other words, every set of treacherous terms has an $\mathcal{I}$-independent treacherous subset.)*

For $\mathcal{I}$-independent substitutions we can show the following:

Proposition 5 *Let $\theta = [x_1 \leftarrow s_1, \ldots, x_n \leftarrow s_n]$ be an $\mathcal{I}$-independent substitution assigning non-public terms s_i to variables x_i, $1 \leq i \leq n$. Then θ unifies two public terms t_1 and t_2 if and only if $t_1 = t_2$.*

Proof: Assume $t_1 \neq t_2$ and let η be an idempotent mgu of t_1 and t_2. We can assume without loss of generality that $\mathcal{V}ar(t_1) \cup \mathcal{V}ar(t_2) \subseteq \{x_1, \ldots, x_n\}$. Then there must be a variable x_i and a public term t such that $\eta(x_i) = t$ and t does not contain x_i. But this means that $s_i \in I(s_1, \ldots, s_{i-1}, s_{i+1}, \ldots, s_n)$ contradicting the assumption that θ is $\mathcal{I}$-independent. □

Proposition 6 *Let $l \rightarrow r$ be a rewrite rule, s be a term such that $\mathcal{P}os(l) \subseteq \mathcal{P}os(s)$ and θ be a ground substitution such that $\theta(s)$ is an instance of l. Then either s is an instance of l or there are distinct subterms s_1 and s_2 of s such that $\theta(s_1) = \theta(s_2)$.*

Proof: Suppose that s is not an instance of l and $\theta(s)$ is an instance of l. Since $\mathcal{P}os(l) \subseteq \mathcal{P}os(s)$, all the variable positions of l must also belong to $\mathcal{P}os(s)$. Since s is not an instance of l, it must be that some variable $v \in \mathcal{V}ar(l)$ has two distinct occurrences at positions p_1 and p_2 of l such that $s|_{p_1} \neq s|_{p_2}$. □

Corollary 2 *Let $l \rightarrow r$ be a rewrite rule, s be a public term such that $\mathcal{P}os(l) \subseteq \mathcal{P}os(s)$ and θ be an I-independent ground substitution such that $\theta(s)$ is an instance of l. Then s is an instance of l.*

Proof: By Proposition 6 there must be distinct subterms s_1 and s_2 of s such that θ is a unifier of s_1 and s_2. But by Proposition 5 s_1 and s_2 must be identical which is a contradiction. □

Proposition 7 *Let S be an $\mathcal{I}$-independent set of ground terms and t any given term. Then the problem of checking whether t has an instance in $\mathcal{I}(S)$ is in* NTIME($|t| + |S|$) *where $|S|$ =* sum of the sizes of terms in S.

Proof: We represent the given set S of ground terms as a (not necessarily rooted) dag $G = (V, E)$; let $V = \{n_1, \ldots, n_l\}$. We define the following mapping with each node in V:

$$\begin{array}{rcll} \nu(n_i) & = & (n_i, 1) & \text{if the term at } n_i \text{ is in } S, \\ & = & (n_i, 0) & \text{otherwise.} \end{array}$$

Each such pair (n_i, b) will be seen as a state of a tree automaton A; we also add a distinguished state q_{acc}, which will be the only accepting state of A. For all nodes n_j, if $f^{(l)}$ is the symbol at the node and $n_{j_1}, \ldots n_{j_l}$ are the nodes corresponding to the ordered arguments of f (not necessarily distinct), then we form a transition rule of A:

$$f(\nu(n_{j_1}), \ldots, \nu(n_{j_l})) \rightarrow \nu(n_j).$$

If $\nu(n_j) = (n_j, 1)$, then we also form the rule:

$$f(\nu(n_{j_1}), \ldots, \nu(n_{j_l})) \rightarrow q_{acc}.$$

Finally, for all public symbols g, we add the transition rule:

$$g(q_{acc}, \ldots, q_{acc}) \rightarrow q_{acc}.$$

The size of the automaton A is obviously linear in $|S|$. The automaton is non-deterministic, but it is easily checked that every term in $\mathcal{I}(S)$ has a unique accepting run.

Now consider the problem of checking whether a given term t has an instance in $\mathcal{I}(S)$. If $p_1, \ldots, p_n$ are the variable positions of t, then we guess the states at each position, say $q_1, \ldots, q_n$ respectively;

1. we have then to verify that this state assignment can be completed into an accepting run for t; *and*
2. for each variable $x \in \mathcal{V}ar(t)$, if $p_{x_1}, \ldots, p_{x_j}$ are the positions where it occurs and $q_{x_1}, \ldots, q_{x_j}$ the corresponding states, then check whether there is a common term t' that all these states "inhabit" — i.e., each state appears at the root of a run of A on the term t'.

Checking this latter requirement (although EXPTIME-hard, in general) is very easy in our case: the only way that $q_{x_1}, \ldots, q_{x_j}$ can appear at the roots of runs for the same term, is if one of the following holds:

(a) they are all the same, or
(b) $\{q_{x_1}, \ldots, q_{x_j}\} = \{q_{acc}, (n, 1)\}$ for some $n \in V$. □

Another way of stating the above conditions (a) and (b), is as follows: for each variable $x \in \mathcal{V}ar(t)$, if $p_{x_1}, \ldots, p_{x_j}$ are the positions where it occurs and $q_{x_1}, \ldots, q_{x_j}$ the corresponding states, then $\{q_{x_1}, \ldots, q_{x_j}\}$ is:

(a') either $\{q_{acc}\}$;
(b') or $\{\nu(n_i)\}$ for some $n_i \in V$;
(c') or $\{q_{acc}, (n, 1)\}$ for some $n \in V$.

This enables us to formulate the following NP-algorithm: for each variable that occurs more than once, guess which of the conditions (a'), (b') or (c') will hold. If (a') then replace x with a public constant c; if (b') or (c') then replace x with the term corresponding to the node. Finally, a linear term s has an instance in $\mathcal{I}(S)$ if and only if s matches with a term in S, or $s = f(s_1, \ldots, s_m)$, where f public, and each $s_i, 1 \leq i \leq m$, has an instance in $\mathcal{I}(S)$.

The deterministic version of this algorithm (i.e., exhaustive search instead of guessing) has time complexity $O(3^k\,(|t| + |S|))$ where k is the number of variables that occur more than once. Thus we have a polynomial time algorithm for the case where this number k is fixed in advance.

Proposition 8 *Let S be an $\mathcal{I}$-independent set of ground terms and t any given term. Then checking whether t has an instance in $\mathcal{I}(S)$ can be done in time $O(3^k\,(|t| + |S|))$ where k is is the number of variables occurring more than once in the term t.*

7.2 A Procedure for the General Cap Problem

We propose an inference rule and a saturation procedure in order to derive the secret m from a given set S of non-public terms. The procedure can be shown to terminate for

all convergent term rewrite systems. It is not complete in general; however, completeness can be shown for dwindling TRSs and p-strong TRSs. And in the dwindling case, the algorithm will be shown to run in NP time.

Let $\mathcal{F}Pos(t)$ be the set of non-variable positions in any term t: $\mathcal{F}Pos(t) = \{\, p \mid p \in \mathcal{P}os(t),\ t|_p \text{ is not a variable} \,\}$. In the proof details below, we shall be denoting by '$\preceq$' the subterm ordering on terms, as well as the prefix ordering on the positions on a term, interchangeably.

The inference rule is as follows:

$$\frac{S \uplus \{s\} \qquad (l,p)}{S \cup \{s,\ \sigma(r)\}} \quad \begin{array}{l}\text{where } (l \to r) \in R,\ \sigma = mgu(s =^? l|_p),\ p \in \mathcal{F}Pos(l),\ \sigma(r) \prec s,\\ \text{and } \sigma(l) \text{ has an instance in } \mathcal{I}(S).\end{array}$$

The set S is said to be *saturated* iff it doesn't grow under any application of this inference rule.

Lemma 4 *Any set of private terms S can be saturated in finitely many steps.*

Proof: The easiest way to see this is to view the inference step as a form of ordered rewriting using the rewrite rules $R' = \{\, (l|_p \to r) \mid (l \to r) \in R,\ p \in \mathcal{F}Pos(l) \,\}$. Since each term has only finitely many descendants modulo R', the saturation process cannot lead to an infinite set of terms. □

Clearly S is treacherous if the saturated set contains m.

The incompleteness of the saturation procedure given above, for general TRS and arbitrarily chosen simplification orderings $\succ$, can be seen from the following example. Consider the following convergent TRS:

$$f(g(x)) \to h(x), \qquad f(h(x)) \to x,$$

where all functions are public, and $\succ$ is the simplification ordering over the symbol precedence $f \succ h \succ g$. Now $\{g(m)\}$ is treacherous since $f(f(g(m))) \to^! m$. But the set $\{g(m)\}$ is already saturated: no inference step can be applied because $h(m) \succ g(m)$.

7.3 An NP-Decision Procedure for dwindling TRS

Proposition 9 *The general cap problem is in NP for any dwindling (convergent) term rewrite system R.*

Proof: The proof uses the following two lemmas (notation of Section 2):

Lemma 5 *Let R be a (convergent) dwindling TRS and let S be a saturated set of private terms. Then S is treacherous if and only if $m \in S$.*

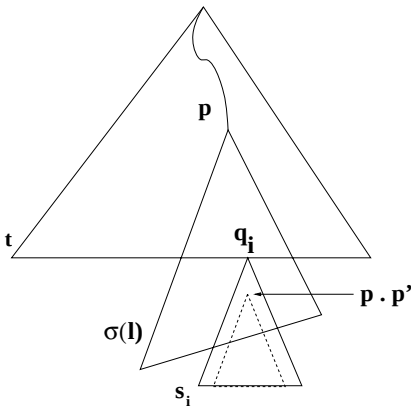
Proof: Assume the contrary. Let S be a saturated set of private terms that is treacherous but does not contain m. Let t' be a $\succ$-minimal term in $\mathcal{I}(S)$ whose irreducible normal form is m. By Proposition 4 there must be a cap $t(\diamond_1, \ldots, \diamond_k)$ and an $\mathcal{I}$-independent

substitution $\theta = [\diamond_1 \leftarrow s_1, \ldots, \diamond_k \leftarrow s_k]$, whose range is a subset of S, such that $t' = \theta(t)$. Suppose t' is reducible by a rule $l \rightarrow r$, and $t'|_p = \sigma(l)$ for some substitution σ. Let $p_1, \ldots, p_n$ be the variable positions of l, and let $\pi_i = p \cdot p_i$ for $i = 1, \ldots, n$.

Now $l \rightarrow r$ is a dwindling rule, i.e., $r = l|_{p'}$ for some position p'; there are two cases to be considered.

(i) $p \cdot p'$ is a position in t. Then $t[p \leftarrow t|_{p \cdot p'}]$ is in $\mathcal{I}(S)$, which contradicts the minimality of t'.

(ii) There is a variable $\diamond_i$ at some position q_i in t such that $q_i \prec p \cdot p'$; hence s_i unifies with a subterm of l.



Now, since S is assumed to be saturated, $\sigma(r)$ has to be already in S. Thus $t'[p \leftarrow \sigma(r)]$ is in $\mathcal{I}(S)$ too, which is a contradiction. □

In the light of the above proof, we can modify the inference rule for the dwindling case to:

$$\frac{S \uplus \{s\} \qquad (l, p)}{S \cup \{s,\ \sigma(r)\}}$$

where $(l \rightarrow r) \in R$, $\sigma = mgu(s =^? l|_p)$, $p \in \mathcal{F}Pos(l)$, r is a proper subterm of $l|_p$, and $\sigma(l)$ has an instance in the $\mathcal{I}$-closure $\mathcal{I}(S)$.

Let $||R|| = \sum_{(l_i \rightarrow r_i) \in R} |l_i|$. We then have:

Lemma 6 *The saturation of any given set of private terms S can be done in non-deterministic time $\phi(|S|, ||R||)$, where ϕ is a polynomial with two arguments.*

Proof: Every term added under an inference is a proper subterm of some term in S; by Proposition 7 each inference step can be performed in NP time. □

For the special case where the number of variables that occur more than once in the left-hand sides of R is fixed, we can get a polynomial-time algorithm by Proposition 8.

Proposition 10 *Let k be a fixed natural integer, and R a dwindling TRS such that, for each $l \rightarrow r \in R$ the number of variables in $Var(l) \setminus Var(r)$ that occur more than once in l is less than k. Then, the general cap problem over R, and a given set of private terms S, is decidable in polynomial time over $||R||$ and $|S|$.*

8 Δ-Strong Intruder Theories

Let R_0 be any given convergent intruder TRS. An n-ary public symbol f is said to be *transparent for* R_0, or R_0-*transparent*, if and only if, for all $x_1, \ldots, x_n$, there exist cap-terms $t_1(\diamond), \ldots, t_n(\diamond)$ such that $t_i[\diamond \leftarrow f(x_1, \ldots, x_n)] \rightarrow^*_{R_0} x_i$, for every $1 \leq i \leq n$. For instance, the public function p ("pair") is transparent for the TRS: $\pi_1(p(x,y)) \rightarrow x,\ \pi_2(p(x,y)) \rightarrow y$, where π_1 and π_2 are public.

It is clear that if the general cap problem is decidable for R_0, then so is checking R_0-transparency. We shall consider public constants to be transparent for any intruder system R_0. A public function symbol is R_0-*resistant* (or simply *resistant* if R_0 is clear from the context) iff it is not R_0-transparent. Private functions will be considered to be resistant, for any intruder system R_0. A term is said to be R_0-resistant (resp. R_0-transparent) iff its top-symbol is so.

Let R be any convergent intruder TRS, and $\succ$ a simplification ordering containing R. (Note: the notation '$\succ$' for the term ordering should not cause any confusion with the prefix-ordering for the positions on terms, since '$\succ$' is a simplification ordering.) We assume that $\succ$ is a precedence based (lpo or rpo like) ordering that satisfies the *block-ordering property: every private symbol is higher than every public symbol under* $\succ$. We shall denote by Δ a subsystem consisting of (some of the) dwindling rules in R. A rewrite rule $l \rightarrow r$ is said to be Δ-strong, wrt the simplification ordering $\succ$, if and only if every Δ-resistant subterm of l is greater than r wrt $\succ$. The intruder TRS R is said to be Δ-strong wrt $\succ$ if and only if every rule in $R \setminus \Delta$ is Δ-strong wrt $\succ$.

Lemma 7 *Let R be a convergent intruder TRS, Δ a convergent dwindling subsystem of R, and suppose R is Δ-strong wrt a simplification ordering $\succ$ satisfying the block-ordering property, total on ground terms, and containing R. Then, any set S of private terms that is saturated (for the inference rule of Section 7.2), is R-treacherous if and only if $m \in S$.*

Proof: We extend our proof of Lemma 5. Let S be a saturated set of private terms that is treacherous, but does not contain m. Let t' be a $\succ$-minimal term in $\mathcal{I}(S)$ whose normal form is m. By Proposition 4, there must be a cap $t(\diamond_1, \ldots, \diamond_k)$ and an $\mathcal{I}$-independent substitution $\theta = [\diamond_1 \leftarrow s_1, \ldots, \diamond_k \leftarrow s_k]$, whose range is an $\mathcal{I}$-independent subset of S (by definition), such that $t' = \theta(t)$. We can also assume that *none* of the s_i's has a Δ-transparent root symbol. Suppose t' is reducible by a rule $l \rightarrow r$, and $t'|_p = \sigma(l)$ for some substitution σ. Let $p_1, \ldots, p_n$ be the variable positions of l, and let $\pi_i = p \cdot p_i$ for $i = 1, \ldots, n$. Now two cases have to be considered:

(i) $l \rightarrow r$ is a dwindling rule, i.e., $r = l|_{p'}$ for some position p': here we conclude exactly as in the proof of Lemma 5.

(ii) $l \rightarrow r$ is a Δ-strong rule: If all the positions π_j are positions in t, we are in the case where t itself is reducible by $l \rightarrow r$ by Corollary 2, contradicting the minimality of t'. So we may assume $\exists j'$ such that $\pi_{j'}$ is not a position in t. Hence there must be a variable $\diamond_i$ at some position q_i in t such that $q_i \prec \pi_{j'} = p \cdot p_{j'}$. Let $q_i = p \cdot q'_i$. Thus $s_i = t'|_{q_i} = \sigma(l|_{q'_i})$ and $q'_i \prec p_{j'}$. But then $l|_{q'_i}$ is a non-ground, non-variable term, and its root must be Δ-resistant, since it unifies with s_i. Since R is Δ-strong, $r \prec l|_{q'_i}$ and thus $\sigma(r) \prec \sigma(l|_{q'_i}) = s_i$. But S is assumed saturated, so $\sigma(r)$ has to be already in S. Thus $t'[p \leftarrow \sigma(r)]$ is in $\mathcal{I}(S)$ too, which is a contradiction. $\square$

Remark 1. It is important to observe that the block-ordering property is essential in step ii) of the proof above; consider the intruder TRS $\{h(f(x)) \to g(x),\ h(g(g(x))) \to x\}$ where the functions h and f are public and g is private, with the ordering $f \succ g \succ h$. The set $\{g(m)\}$ is saturated, but $h(h(f(\diamond)))$ is a cap that works, since $h(h(f(g(m)))) \to^! m$. Note that this cap is reducible, and $h(g(\diamond))$ is its normal form, but this latter term *cannot* be a 'legal cap' since it contains the non-public symbol g. Such anomalies will not arise, of course, if the ordering is assumed to satisfy the block-ordering property.

We thus get

Proposition 11 *The following problem is decidable:*

Instance: *A convergent TRS R over the intruder repertoire, Δ a dwindling, convergent subsystem of R, a simplification block-ordering $\succ$ wrt which R is Δ-strong, a free constant m, and a finite set S of irreducible non-public ground terms, at least one of which contains m.*

Question: *Is there a cap-term $t(\diamond_1, \cdots, \diamond_k)$ such that $t[\diamond_1 \leftarrow s_1, \cdots, \diamond_k \leftarrow s_k]$, with the $s_i \in S$ (not necessarily all distinct), can be R-reduced to m?*

Applications. (i) One can handle the cap problem for homomorphic encryption (i.e., 'encryption' e distributes over 'pair'), by the Δ-strong approach, with the following convergent TRS R; the rules to the left form Δ; d, e are Δ-resistant:

$$\begin{array}{rcl} \pi_1(pair(x,y)) & \to & x \\ \pi_2(pair(x,y)) & \to & y \end{array} \qquad \begin{array}{rcl} d(e(x,y),y) & \to & x \\ e(d(x,y),y) & \to & x \\ e(pair(x,y),z) & \to & pair(e(x,z),e(y,z)) \\ d(pair(x,y),z) & \to & pair(d(x,z),d(y,z)) \end{array}$$

Related results were obtained in [12], with a more complex proof (but with a polynomial time algorithm).

Homomorphic encryption and signature have several applications, such as e-voting, auction, and private information retrieval.

(ii) The blind signature protocol can be modeled by rewrite rules of the form

$$\mathcal{U}(S_A(\mathcal{B}_A(x,y)),y) \to S_A(x),$$

where $\mathcal{B}_A$ is the blinding function (of B wrt to signer A), S_A is the signing function of A, and $\mathcal{U}$ is the unblinding function. Such systems are covered by the Δ-strong approach, by setting $\mathcal{B}_A \succ S_A$, for every signer A. One can also handle Block-Cipher related theories, such as the one obtained by adding the rule $split(e(pair(x,y),z)) \to e(x,z)$ to the dwindling system Δ of the previous example (i).

Remark 2. (i) The preceding lemma and proposition seem to remain valid, if Δ is replaced by any convergent subsystem $R_0 \subset R$ for which the general cap problem is decidable.

(ii) No polynomial upper bound can be given for the number of terms added to S, under saturation, in the Δ-strong case (unlike in Lemma 6 for the dwindling case). We show in Appendix-II that the complexity is non-primitive recursive.

8.1 $\omega\Delta$-strong Intruder Theories

As before, R denotes a given convergent intruder TRS, $\succ$ a simplification ordering containing R, and Δ some given, dwindling and convergent subsystem of R. For any rule $l \to r \in R$, let $\mu(l)$ stand for the set of $\succ$-minimal subterms of l that are Δ-resistant, in the sense defined earlier. We define then a notion on R that is weaker than that of Δ-strong considered earlier; this notion, referred to and denoted as *$\omega\Delta$-strong*, is defined as follows ('ω' stands for 'weak'):

Definition 1 *A rule $l \to r \in R$ is said to be $\omega\Delta$-strong wrt $\succ$, if and only if there* exists *a position p on l such that $l|_p \in \mu(l)$ and $l|_p > r$. The system R is $\omega\Delta$-strong wrt $\succ$, if and only if every rule in R is $\omega\Delta$-strong wrt $\succ$.*

We propose to show that the result of Lemma 7 continues to hold under the weaker assumption that R is $\omega\Delta$-strong wrt $\succ$, again under the assumption that the ordering $\succ$ satisfies the block-ordering property defined above. In what follows, $\mathcal{S}ub(S)$ will stand for the set of all subterms of the set of terms S.

Lemma 8 *Let S be an $\mathcal{I}$-independent set of non-public ground terms such that every term in S has a Δ-resistant symbol at its root. Let $l \to r$ be a rule in R, σ a ground substitution, and $p \in \mathcal{F}Pos(l)$ such that $l|_p \in \mu(l)$ and $l|_p \succ r$. If $\sigma(l|_p) \in \mathcal{I}(S)$ and $\sigma(r) \notin \mathcal{I}(S)$, then $\sigma(l|_p) \in \mathcal{S}ub(S)$.*

Proof: Suppose $\sigma(l|_p) \in \mathcal{I}(S)$, and $\sigma(r) \notin \mathcal{I}(S)$. We must then have:

(i) either r contains a private symbol,

(ii) or $\mathcal{R}an(\sigma|_{\mathcal{V}ar(r)}) \not\subseteq \mathcal{I}(S)$,

or both. If r contains a private symbol then so should $l|_p$. Since $l|_p$ is a $\succ$-minimal Δ-resistant subterm of l, this private symbol must be at the root of $l|_p$. Now, by definition, the root symbol of a term in $\mathcal{I}(S)$ that is not already in S cannot be private; hence $\sigma(l|_p)$ cannot be in $\mathcal{I}(S)$, unless already in S.

If r does not contain a private symbol, then there must be a variable v in $\mathcal{V}ar(r)$ and hence in $\mathcal{V}ar(l|_p)$ such that $\sigma(v) \notin \mathcal{I}(S)$; thus $\sigma(v)$ itself is not in $\mathcal{I}(S)$, but a superterm of it, namely $\sigma(l|_p)$, is in $\mathcal{I}(S)$. Therefore $\sigma(v)$ must be a *proper* subterm of some term in S. Let $s \in S$ be a term that $\sigma(v)$ is a proper subterm of. Now, $l|_p$ is a minimal Δ-resistant subterm of l, so all function symbols of $l|_p$, except at its root, must be Δ-transparent; but the root of s is Δ-resistant by our assumption above, so it must be that $\sigma(l|_p)$ is a subterm of s; thus $\sigma(l|_p) \in \mathcal{S}ub(S)$. □

Lemma 9 *Let R be a convergent intruder TRS, $\succ$ a simplification ordering total on ground terms that satisfies the block-ordering property and contains R, and Δ a convergent dwindling subsystem of R. Suppose R is $\omega\Delta$-strong wrt $\succ$. Then, any saturated set S of private terms is R-treacherous if and only if $m \in S$.*

Proof: Let S be as assumed in the hypothesis, and suppose S does not contain m. Let t' be a $\succ$-minimal term in $\mathcal{I}(S)$ whose normal form is m. Then there must be a cap $t(\diamond_1, \ldots, \diamond_k)$ and an $\mathcal{I}$-independent substitution $\theta = [\diamond_1 \leftarrow s_1, \ldots, \diamond_k \leftarrow s_k]$, whose range

is an $\mathcal{I}$-independent subset of S, such that $t' = \theta(t)$. Let $\widehat{S} = \{s_1, \ldots, s_k\}$. We can assume, without loss of generality, that the terms in $\widehat{S}$ are R-irreducible and also that none of the s_i's has a Δ-transparent root symbol. Suppose t' is reducible by a rule $l \to r$, and $t'|_p = \sigma(l)$ for some substitution σ. We can assume that $\sigma(r) \notin \mathcal{I}(S)$, for otherwise $t'[p \leftarrow \sigma(r)] \in \mathcal{I}(S)$ contradicting the minimality of t'. We have two cases to consider:

(i) $l \to r$ is a dwindling rule: *we conclude as earlier.*

(ii) $l \to r$ is an $\omega\Delta$-strong rule: Let $q \in \mathcal{FP}os(l)$ such that $l|_q \in \mu(l)$ and $l|_q \succ r$. Since $\sigma(r) \notin \mathcal{I}(S)$, it must be that $\sigma(l|_q)$ is a subterm of some term in $\widehat{S}$, by the preceding lemma. Now, the elements of $\widehat{S}$ are R-irreducible, so the root position of $\sigma(l)$ is in the "cap" part of t'; thus, there must be a position $\epsilon \prec q' \preceq q$ in $\mathcal{FP}os(l)$ such that $\sigma(l|_{q'}) = s$, for some $s \in \widehat{S}$; since $l|_q \succ r$, we get $s = \sigma(l|_{q'}) \succ \sigma(r)$. But S is assumed to be saturated, so $\sigma(r)$ must be in S too; contradiction. □

9 Related Works, Conclusion

In [14], the authors have studied intruder theories given by convergent public-collapsing systems. They give an NP-decision procedure for protocol insecurity in the case of an active intruder. In [1], the authors present an algorithm for the general cap problem for an intruder given by a convergent *dwindling* rewrite system. They in fact considered the more general *static equivalence* problem, and their algorithm was proved to be polynomial when the size of the rewrite system is *fixed*. This work has been extended by [5] to the insecurity problem for active intruders; but the author does not give any complexity result. We have shown that the extension of even the (single hole) cap problem to the slightly more general class of optimally reducing, (convergent) and linear TRS leads to undecidability; but our decidability result for such systems in the string rewriting case, improves upon the results of Book and Otto [7]. The NP-complexity bound established for the general cap problem wrt dwindling TRS, can be seen as adding precision to some results of [1]; actually, our complexity result of Proposition 10, for the general cap problem over such systems, is stronger than the corresponding result of [1]. We have also given an algorithm for the general cap problem for a class of intruder theories not considered in [14, 1, 5], namely the Δ-*strong* one, thereby deriving a new security result for passive intruders. In [12], a deduction problem analogous to the general cap problem is investigated, by using specific deduction rules for encryption and pairs (unlike ours), and it is unclear how the results can be compared.

The decidability results derived in this paper cover several theories of interest for security protocols. It would be of interest to extend them to AC-rewrite systems, in order to capture important theories comprising AC-operators (e.g., abelian groups). It would be important too, to try to lift our results to cover the case of active intruders, by integrating constraint solving and semantic unification algorithms. In this regard, it must be noted that the cap problem wrt an intruder theory R, and the problem of unification modulo R, behave in an unrelated manner, in general: indeed, for any given ground term t, the set of its capped versions, wrt any given rewrite system R, is a regular tree language; thus, by Theorem 5.1 of [13], it follows that the cap problem wrt R is decidable if the intruder theory R is linear and (semi-)monadic; unfortunately though, the unification problem is undecidable for linear and monadic TRS, as is shown in Appendix-I. This is in marked

contrast to optimally reducing systems R, where R-unification is decidable and finitary, but the cap problem is undecidable, as we saw in Section 6.

References

[1] M. Abadi, V. Cortier. Deciding knowledge in security protocols under equational theories. In *Theor. Comput. Sci.*, 367(1-2):2–32, 2006.

[2] R. Amadio, D. Lugiez, and V. Vanackère. On the symbolic reduction of processes with cryptographic functions. *Theor. Comput. Sci.*, 290(1):695–740, 2003.

[3] S. Anantharaman, P. Narendran, M. Rusinowitch. Unification modulo ACUI plus Distributivity Axioms. In *Journal of Automated Reasoning* 33: 1–28, 2004.

[4] A. Armando, L. Compagna. SATMC: a SAT-based Model Checker for Security Protocols, *Proc. of JELIA 2004*, LNCS 3229, pp. 730–733, Springer-Verlag, 2004.

[5] M. Baudet. Deciding security of protocols against off-line guessing attacks. In *Proc. of ACM Conference on Computer and Communications Security*, 2005, pp. 16-25.

[6] R.V. Book, M. Jantzen, C. Wrathall. Monadic thue systems. In *Theor. Comput. Sci.* 19:231–251, 1982.

[7] R.V. Book, F. Otto. The verifiability of two-party protocols. In *EUROCRYPT,* pages 254–260, 1985.

[8] R.V. Book, F. Otto. *String-Rewriting Systems.* Springer-Verlag, 1993.

[9] Y. Chevalier and L. Vigneron. A Tool for Lazy Verification of Security Protocols. In *Proceedings of the Automated Software Engineering Conference (ASE'01).* IEEE Computer Society Press, 2001.

[10] H. Comon, M. Dauchet, R. Gilleron, F. Jacquemard, D. Lugiez, S. Tison, M. Tommasi. Tree Automata Techniques and Applications. Available at: `http://www.grappa.univ-lille3.fr/tata/`

[11] H. Comon-Lundh, S. Delaune. The finite variant property: how to get rid of some algebraic properties. In *Proc. of RTA'05* (Jürgen Giesl, ed.), LNCS 3467, pages 294–307. Springer-Verlag, 2005.

[12] H. Comon-Lundh, R. Treinen. Easy Intruder Deductions. Verification: Theory and Practice In *Lecture Notes in Computer Science 2772*, pages 225–242, Springer-Verlag, 2003.

[13] J-L. Coquidé, M. Dauchet, R. Gilleron, S. Vágvölgyi. Bottom-up tree pushdown automata: Classification and connection with rewrite systems. In *Theor. Comput. Sci.* 127(1):69–98, 1994.

[14] S. Delaune, F. Jacquemard. A decision procedure for the verification of security protocols with explicit destructors. In *Proc. of ACM Conference on Computer and Communications Security,* 2004, pp. 278–287.

[15] D. Dolev, A.C. Yao. On the security of public key protocols. In *IEEE Transactions on Information Theory* 29(2):198–207, 1983.

[16] N.A. Durgin, P.D. Lincoln, J.G. Mitchell, A. Scedrov. Multiset rewriting and the complexity of bounded security protocols. In *Journal of Computer Security* 12(1):677–722, 2004.

[17] R. Gilleron, S. Tison. Regular tree languages and rewrite systems, In *Fundamenta Informaticae* 24, 157–176, 1995.

[18] G.P. Huet. Confluent reductions: Abstract properties and applications to term rewriting systems. In *Journal of the ACM* 27(4):797–821, 1980.

[19] M. Minsky. *Computation: Finite and Infinite Machines.* Prentice-Hall International, London, 1972.

[20] P. Narendran, F. Pfenning, R. Statman. On the unification problem for Cartesian Closed Categories. In *Journal of Symbolic Logic* 62 (2), June 97, 636–647.

[21] M. Nesi, G. Rucci. Formalizing and analyzing the Needham-Schroeder symmetric-key protocol by rewriting. In *Electr. Notes Theor. Comput. Sci.* 135(1):95–114, 2005.

[22] M. Rusinowitch, M. Turuani. Protocol insecurity with a finite number of sessions, composed keys is NP-complete. In *Theor. Comput. Sci.* 1-3(299):451–475, 2003.

[23] C. Weidenbach. Towards an automatic analysis of security protocols. In Proc. of *16th International Conference on Automated Deduction, CADE-16*, LNAI 1632 (H. Ganzinger, ed.), Springer-Verlag, pages 378–382, 1999.

Appendix-I

We show here that there is a linear, monadic, convergent TRS for which the unification problem is undecidable. (Note: A TRS R is said to be monadic, iff for every rule $l \to r \in R$, we have $depth(l) \geq 1$ and $depth(r) \leq 1$.) The reduction is from a restricted version of the modified Post Correspondence Problem (MPCP).

Let $\Sigma = \{a, b\}$, and let $P = \{(x_i, y_i) \mid i = 1, \ldots, n\} \subseteq \Sigma^+ \times \Sigma^+$ be a finite sequence of non-empty strings over Σ such that the following restricted version of the Modified Post Correspondence Problem (MPCP) is undecidable:

Instance:	A non-empty string $\alpha \in \Sigma^+$.
Question:	Do there exist indices $i_1, \ldots, i_k \in \{1, \ldots, n\}$ such that $\alpha x_{i_1} x_{i_2} \ldots x_{i_k} = y_{i_1} y_{i_2} \ldots y_{i_k}$?

For a string w over Σ, let $\widetilde{w}(x)$ denote the term formed by treating a and b as unary function symbols and the concatenation operator as function composition; more precisely, we set:

$$\widetilde{\lambda}(x) = x, \quad \widetilde{au}(x) = a(\widetilde{u}(x)), \quad \widetilde{bu}(x) = b(\widetilde{u}(x)).$$

Let f be a ternary function and $g_1, \ldots, g_n$ be unary functions. We construct a linear, monadic TRS, consisting of the following rules:

$$f(\widetilde{x_i}(u),\, g_i(v),\, \widetilde{y_i}(w)) \to f(u,\, v,\, w)$$

for every pair (x_i, y_i) of the MPCP. (The role played here by the g_i is one of ensuring that the rewrite system has no critical pairs.)

It is not hard then to see that the unification problem

$$\{f(X,\, \widetilde{\alpha}(X),\, Y) \;=^?\; f(c,\, c,\, c)\}$$

has a solution if and only if the instance of the restricted MPCP above has a solution.

Appendix-II

We show here that the complexity of the general cap problem is non-primitive recursive, for (general) Δ-strong intruder theories. The proof is based on the following lines of reasoning. The starting point is the following observation of Petr Jančar, where $\mathcal{A}(n)$ is a non-primitive recursive function on natural integers:

> "The problem to decide, given a 2-counter machine C and a natural number n, whether C halts on zero input in $\mathcal{A}(n)$ steps is non-primitive recursive".

in "*Nonprimitive recursive complexity and undecidability for Petri net equivalences*" (Theor. Comp. Science, 256(1-2):23-30, 2001; Proposition 9, Section 4).

Let then C be an arbitrarily given 2-counter machine, with $L + 1$ instructions; to each instruction of label i, $0 \leq i \leq L$, is associated a state denoted as q_i, which will be seen as a unary function on $\mathbb{N}$ (as in Section 6). And consider the following convergent TRS R_0,

where 0 stands for the natural number 0, s for the successor function on $\mathbb{N}$, and p stands for the predecessor function defined as usual:

$$\begin{aligned} f(0,x) &\to s(x) \\ f(s(x),0) &\to f(x,s(0)) \\ f(s(x),s(y)) &\to f(x,f(s(x),y)) \\ p(s(x)) &\to x \\ h(g(x,u,v,w)) &\to r(f(x,x),u,v,w) \\ d(q_i(x)) &\to x, \quad 0 \le i \le L. \end{aligned}$$

The function f obviously encodes the usual Ackermann function (on two arguments over $\mathbb{N}$). The symbol h plays no specific role, other than ensuring that a term with top symbol g is R_0-irreducible. The last set of rules, plus the fourth, constitute a dwindling convergent sub-TRS Δ, wrt which the q_i's and s are Δ-transparent; the other symbols are all Δ-resistant. We then encode the instructions of the given 2-counter machine C by the following set of rewrite rules, where the second and the fourth arguments, under the symbol r in the terms to the left, stand for the values of the two counters of C, respectively (and the l, l' are suitable instruction labels):

Incrementation of counter 1 or 2:

$$h(r(s(u),x,q_l(z),y)) \to r(u,s(x),q_{l+1}(z),y),$$
$$h(r(s(u),x,q_l(z),y)) \to r(u,x,q_{l+1}(z),s(y))$$

Conditional decrementation of counter 1 or 2:

$$h(r(s(u),s(x),q_l(z),y)) \to r(u,x,q_{l+1}(z),y),$$
$$h(r(s(u),0,q_l(z),y)) \to r(u,0,q_{l'}(z),y).$$
$$h(r(s(u),x,q_l(z),s(y))) \to r(u,x,q_{l+1}(z),y),$$
$$h(r(s(u),x,q_l(z),0)) \to r(u,x,q_{l'}(z),0).$$

At STOP, release the secret m: $\quad h(r(u,v,q_L(z),w)) \to z.$

Let R denote the intruder theory, formed of all these encoding rules and the rules of the TRS R_0 given above; R is obviously Δ-strong under the *lpo* based on the precedence $0 < q_i < s < p < f < g < r < h$ (where $0 \le i \le L$), and it is also convergent under this simplification ordering.

Finally consider the singleton set $S = \{g(s^n(0),0,q_0(m),0)\}$, where n is some given (fixed) positive integer, and m a given ground constant $\neq 0$. Now, it is not hard to check that this set S is treacherous for R and the 'secret' m, *if and only if* the machine C, with initial counter values both 0, halts under instruction L in exactly $f(n,n)$ steps. □